Trace Interpolation Algorithm Based on Intersection Vehicle Movement Modeling*

Jinwei Shen, Guangtao Xue#
Department of Computer Science & Engineering, Shanghai Jiao Tong University, Shanghai, China
E-mail: xue-gt@cs.sjtu.edu.cn
Received July 9, 2010; revised August 16, 2010; accepted September 20, 2010

Abstract

Real vehicle tracking data play an important role in the research of routing in vehicle sensor networks. Most of the vehicle tracking data, however, were collected periodically and could not meet the requirements of real-time by many applications. Most of the existing trace interpolation algorithms use uniform interpolation methods and have low accuracy problem. From our observation, intersection vehicle status is critical to the vehicle movement. In this paper, we proposed a novel trace interpolation algorithm. Our algorithm used intersection vehicle movement modeling (IVMM) and velocity data mining (VDM) to assist the interpolation process. The algorithm is evaluated with real vehicle GPS data. Results show that our algorithm has much higher accuracy than traditional trace interpolation algorithms.

Keywords: Trace Interpolation, Intersection Vehicle Movement Modeling, Velocity Data Mining, Vehicle Sensor Network

1. Introduction

Vehicular AdHoc Network (VANET) is a special kind of Delay-tolerant Network (DTN). Because of the uncertainty and high mobility of VANET, routing and data sharing in VANET are quite different from MANET. Many papers were published around data delivering/sharing and routing in VANET. Some algorithms are proved upon fully simulation while other algorithms are simulated with real vehicle tracking data. The later would be more persuasive of course. So how to effectively measure the performance of these algorithms depends heavily on the vehicle tracking data.

There are two main kinds of positioning techniques: GPS and cellular positioning technology [1]. Because of the weakness of GPS including long positioning time, bad signal in downtown, high cost of positioning [2]. Cellular positioning technology is used widely in vehicle management. However it also suffers from low accuracy of location (around hectometer [3,4]) and charge by the times of positioning[1]. So most tracking data collected by both positioning technology has accuracy and long-interval problem. Lots of map matching algorithms [5-7] had been proposed to solve the problem of data inaccuracy. However few researches focused on trace interpolation algorithm, which was aimed to solve the long-interval between records problem and to provide a “real time” vehicle trace. Most published works [8] about vehicle trace interpolation use uniform interpolation method. It assumes that the vehicle moves with the same velocity or with a uniform acceleration/deceleration velocity between two consecutive real records. But uniform interpolation method has one obvious problem: it cannot represent the actual vehicle trace, for the vehicle may had a process of deceleration and acceleration or even stops between two consecutive records.

In our observation, some routing and data delivering algorithms [9-11] in VANET uses a special technique which was mostly called “intersection buffering”, this method relies on the underlying feature of vehicle mobility: vehicles tend to emerge at intersections because of the intersection traffic light. Intersections are hot areas of data exchange and delivering.

With the basic idea of introduce IVMM and VDM into trace interpolation. We proposed a novel trace interpolation algorithm. In this algorithm, we fully utilize the information of the history vehicle records and around vehicle records to increase the accuracy of trace interpola-
The rest of this paper is organized as follows. In Section 2, we introduce the IVMM and VDM technique and our trace interpolation algorithm. Section 3 provides the experimental results based on real tracking data. Section 4 includes the conclusion of this paper.

2. Trace Interpolation Algorithm

This paper’s work is based on data collected by 4000 taxis in Shanghai urban setting during several months in 2006, and we have the digital map information of the whole area. Every record in the dataset includes the vehicle’s id, vehicle’s GPS position, velocity information, vehicle direction and timestamp. The interval of the records varies from seconds to minutes.

Table 1 presents us two consecutive vehicle tracking records with some fields omitted.

Since map matching is not what we concerned in this paper, we assume that record $r_1$ was matched to road position $p_1$ on road $C_1C_2$ and record $r_2$ was matched to position $p_2$ on road $C_3C_4$, and the vehicle drives from $p_1$ to $p_2$ through cross $C_2$ and $C_3$ as depicted in Figure 1.

Suppose $t_1$ and $t_2$ has a gap of 1 min, if the required timestamp granularity by application is 1 sec, then trace interpolation is used to get the records or to find out where the vehicle is at each second between $t_1$ and $t_2$. If we know how fast the vehicle drives at each position along the matched roads, it will be easy for us to know where the vehicle is at timestamp $t_x$. So trace interpolation problem could be mapped to velocity finding problem.

In uniform interpolation algorithm, uniform velocity distribution was adopted, which means during the time $t_1$ to $t_2$ the velocity of the vehicle is calculated by $V = \frac{(v_1C_2 + C_2C_3 + C_3p_2)}{(t_2 - t_1)}$. This could hardly match the real case.

Following introduces our trace replay algorithm:

2.1. Intersection Vehicle Movement Modeling

Traditional interpolate algorithm assumes the vehicle move through the intersections at its normal speed without deceleration and stop. In reality, vehicles rarely keep the normal speed at the intersection because of traffic control signals [11]. Most vehicles experience deceleration and acceleration and often wait in line with full stop [12].

There are many different intersection structures in reality, such as signalized, isolated, roundabout, etc. Our intersection velocity model only studies the vehicle movement at the signalized intersection with two crossing paths. However different intersections would still have different traffic light models, and it is hard to precisely build different intersection models to different intersections for we don’t have the traffic light information for each intersection. What’s more, a precise intersection model requires a relatively high volume of traffic density. We did some analysis based on the dataset and found there’s an average of 10 vehicle records near one intersection per 2 min. With this observation, we compromised to build a simplified intersection model for all the signalized intersections. In this model, we had a simplified traffic light model: (1) turn to right is always permitted (2) one of the two directions is fully permitted to go any direction at a time, and we assume the queue of vehicles would not surpass a specified length, which means the vehicle would not stop at a position that is far away from an intersection.

Since most vehicles experience deceleration and acceleration and sometimes wait in line with full stop, in our simplified model, we divide the intersection vehicle velocity model into two categories. As shown in Figure 2. The first one has a full stop while the second one just has a slight deceleration and acceleration.

When a vehicle’s two consecutive records matched on two different roads, the interpolated trace between them would get through intersection. Our task is to find out the intersection status of $C_i$ at any time $t_x$. Thus we will be able to distinguish if a vehicle stops at the intersection $C_i$ at time $t_x$ and how long it stops if it did stops.

We utilize a voting mechanism to decide the status $S_i = \begin{cases} 1, & \text{vertical direction green} \\ 0, & \text{vertical direction red} \end{cases}$ of an intersection $C_i$ at time $t_x$.

First we find the involved (time-space close) record

![Figure 1. Map matched vehicle records.](image)
set \( R = \{ r_1, r_2, \ldots, r_n \} \), \( r_i \) is a record with \(-T_r < t_i - t_x < T_r\) and \(-T_{d1} < d(p_i, C_i) < T_{d1}\), \( T_r \) and \( T_{d1} \) are constant thresholds of involved time range and involved distance range respectively, \( p_i \) is the position of record \( r_i \), \( d(p_i, C_i) \) is the distance of \( p_i \) and cross \( C_i \).

Then each record in \( R \) gives an answer about the status of the intersection, denoted as \( s_i \) with a weight \( w_i \).

\[
\begin{align*}
\begin{cases}
0, & v_i < T_{v_l} \text{ and } l_i = 1 \quad \text{(case1)} \\
1, & v_i > T_{v_h} \text{ and } l_i = 1 \text{ and } d(r_i, C_i) < T_{d2} \quad \text{(case2)} \\
1, & l_i = 0 \quad \text{(case3)}
\end{cases}
\end{align*}
\]

\[
\begin{align*}
\begin{cases}
(T_{t_l} - |t_i - t_x|) / T_{t_l}, & \text{(case1)} \\
(T_{t_l} - |t_i - t_x|) / T_{t_l} \times (T_{d3} - d(r_i, C_i)) / T_{d3} & \text{(case2)} \\
(T_{t_l} - |t_i - t_x|) / T_{t_l} & \text{: (case3)}
\end{cases}
\end{align*}
\]

\[
l_i = \begin{cases} 
1, & \text{ri entering cross } C_i \\
0, & \text{ri leaving cross } C_i
\end{cases}
\]

Case1 is the red light case while case2 and case3 are green light cases. \( T_{v_l} \) and \( T_{v_h} \) are the velocity low and high threshold respectively, \( T_{t_l} \) and \( T_{d3} \) are time and distance threshold respectively. The weight \( w_i \) decreases when \(|t_i - t_x|\) increases.

The formula for \( s_i \) is under the condition \( r_i \) is on vertical direction road, if not, \( s_i \) is flipped.

A simple example of case1: suppose we have two consecutive records \( r_1 \) and \( r_2 \) of vehicle x1 passing intersection \( C_2 \) as shown in Figure 3. If \( r_1 \)'s speed is close to 0, then we could get the information that at time \( t_1 \) traffic light at intersection \( C_2 \) in vertical direction is red while horizontal direction is green.

Finally after all the \( s_i \) and \( w_i \) is calculated, \( S_i \) is given by the following formula.

\[
S_i = \begin{cases} 
1, & \text{if } \sum_{i=1}^{n} s_i \cdot w_i > 0 \\
0, & \text{if } \sum_{i=1}^{n} s_i \cdot w_i \leq 0
\end{cases}
\]

A general deceleration and acceleration process is adopted if \( R \) is empty.

### 2.2. Velocity Data Mining

While intersection modeling solved the problem of intersection interpolation, in positions like center-segment of the road, we however could have no velocity information. Velocity data mining is adopted to improve the interpolation result.

Vehicle’s velocity mainly depends on three factors: (1) road condition and road attribute (speed limitation) (2) nearby vehicle velocity (3) driver habits. Road condition and road attribute were reflected at the overall average velocity of all the vehicle records in history on the specific road segment. Nearby vehicle velocity could be obtained dynamic from space-time-close vehicle records. Unlike road condition/attribute and around vehicle velocity driver habits only depends on the driver itself, history velocity information of the specific vehicle on this road segment could be used to calculate this factor.

Thus to find the most likely velocity of vehicle \( x_1 \) on road \( r_1 \) at time \( t_1 \), we define three dataset (DS) to assist calculation. \( DS_1 \): those records whose road id is \( r_1 \), \( DS_2 \): those records whose road id is \( r_1 \) and timestamp is close to \( t_1 \), \( DS_3 \): those records whose vehicle id is \( x_1 \) and road id is \( r_1 \).

The suggested velocity \( V \) could be represented by the following formula then. \( V_1 \) is the average velocity of the \( i_{th} \) dataset \( DS_i \), \( W_1 \) is the weight of the \( i_{th} \) factor.

\[
V = V_1 \times W_1 + V_2 \times W_2 + V_3 \times W_3
\]

### 2.3. Interpolation

To do interpolation, we first divide the road into three segments, as depicted in Figure 4. We assume the vehicle drives with a uniform acc/dec pattern on the two end-segment \( C_1A \) and \( C_2B \) and a uniform velocity on the center-segment \( AB \).

Then we find out the velocity on the center-segment.
If no velocity information is available on $AB$ during the process of interpolation, then a VDM is used to get the interpolated velocity.

Third we get the status of the nearby intersection by IVMM, different velocity model (fully stop or slight dec/acc) will be adopted to different intersection status.

After the velocity for the whole road have all been set. A general scale $sv$ is set to the velocity to fulfill the equation: $\int_{t_1}^{t_2} s V dt = \text{length}$.

Finally the interpolated relative position on road $C_1C_2$ at $t_i$ could be calculated by $\int_{t_1}^{t_i} s V dt$.

3. Experimental Results

To utilize this dataset to check the accuracy of our algorithm, we picked an area of about $5000 \times 5000 \text{ m}^2$ where traffic density is relatively high, and since map match is not we concerned in this paper, we did the map match as a pre-work for our algorithm with an existing map match algorithm. After the map match, every record locates on the road and knows the path to the next record. We then marked some proportion of the vehicle records as masked (do not take it into calculation) in interpolate process. To get the accuracy of the interpolate algorithm, we only need to compare the interpolated records with the masked records. As described in Table 2, $a_2$ is marked as masked, then the interpolate algorithm will take $a_1$ and $a_3$ as input to get the interpolation result. There will be several new records added between $a_1$ and $a_3$, one of them will have the same timestamp as $t_2$, compare the GPS coordinate and velocity with $a_2$, we got the accuracy.

Figure 5 and Figure 6 are the accuracy comparison results of three different interpolation methods. The first interpolation method is uniform interpolation. The second one is VDM assisted interpolation and the third method is interpolation with IVMM and VDM.

As shown in Figure 5, uniform interpolation has the highest distance error. When the masked data percentage is low, interpolation with VDM has a 10% decrease in distance error, and 50% decrease when both IVMM and VDM are used to assist the interpolation. However the distance error difference gets small as the masked data percentage increases. As we have expected, our IVMM and VDM based interpolation algorithm has higher accuracy advantage over other interpolation algorithms with stronger data set. The velocity error comparison results showed in Figure 6 reaches the same conclusion.

4. Conclusions

In this paper, we proposed a novel trace replay algorithm, which is assisted by IVMM and VDM. Through experiments over real vehicle tracking data collected in Shanghai urban setting, we compared the interpolation accuracy of three different interpolation algorithms. The result shows that our new algorithm has much higher accuracy than existing algorithms. Our algorithm can be easily extended to fit in more complicated intersection models, we believe that with stronger data set support, the accuracy
of our algorithm can be even higher.

5. References


